



## Gradle插件开发秘籍之断点调试（基于IntelliJ）

gradle android java java-web gradle-experimental 风云信步 2017年02月06日发布

Gradle插件开发这件事说大不大说小不小，但是对于有一定体量规模的Java项目来讲，从插件开发入手来思考解决问题，有的时候能找到意想不到的法门。所以说这是一门锦上添花的手艺。

废话少说，相信对于很多Gradle插件开发朋友们，开发过程中很大的一个痛点就是，插件的调试很痛苦，下面我就分享一下在这方面的心得。

### 一、增加运行参数

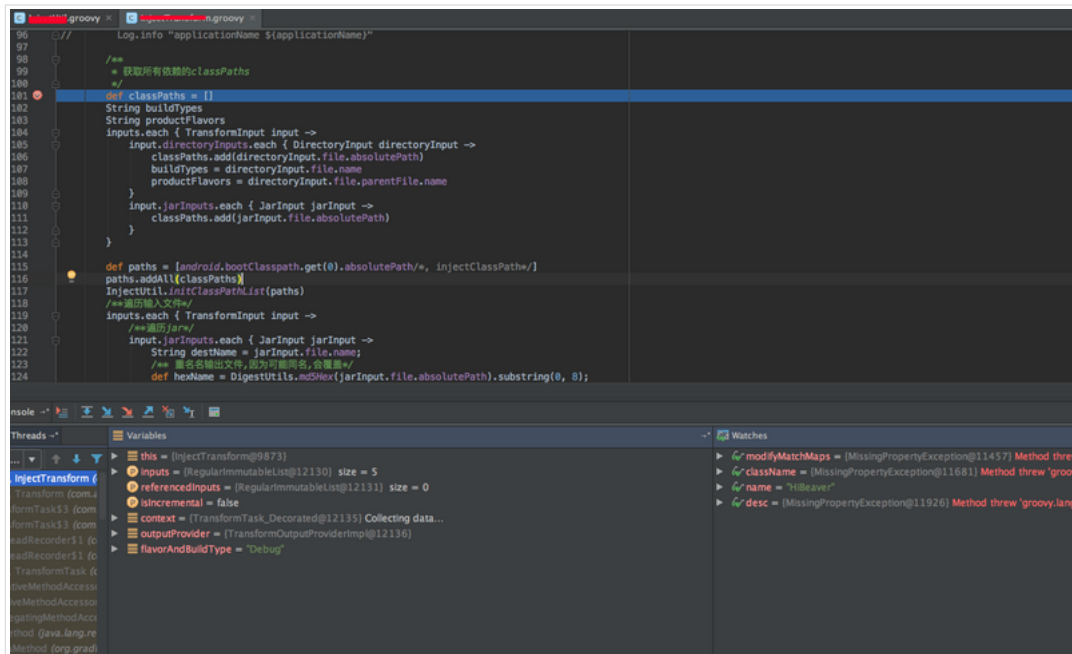
好吧这个很菜鸟，比如很常用参数的是--stacktrace，报错后会打印出堆栈信息

### 二、Log输出控制

这个不多说了，println就可以了，可以自己设置一些分级和flag。

### 三、干货技巧：开启调试模式，attach a debugger!

先上最终效果图：



如上图，是可以正常地设置断点调试和step into和step out、resume等等。

实现办法和原理：

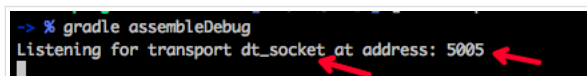
大家都知道Gradle基于Groovy，而Groovy其实就是Java，所以也同样遵循JDWP协议，只是这个技巧似乎在国内的文章中没有人出来分享，我就在此分享给大家。

首先，在你要执行gradle命令的环境中加入如下环境变量（以Linux、MacOS为例）：

```
export GRADLE_OPTS="-Xdebug -Xrunjdpw:transport=dt_socket,server=y,suspend=y,address=5005"
```

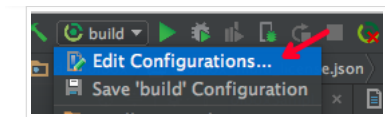
其中，address等号后面的参数就表示debugger要监听的端口，这里我们设置为5005。

然后执行gradle或gradlew任务，会发现如下提示：

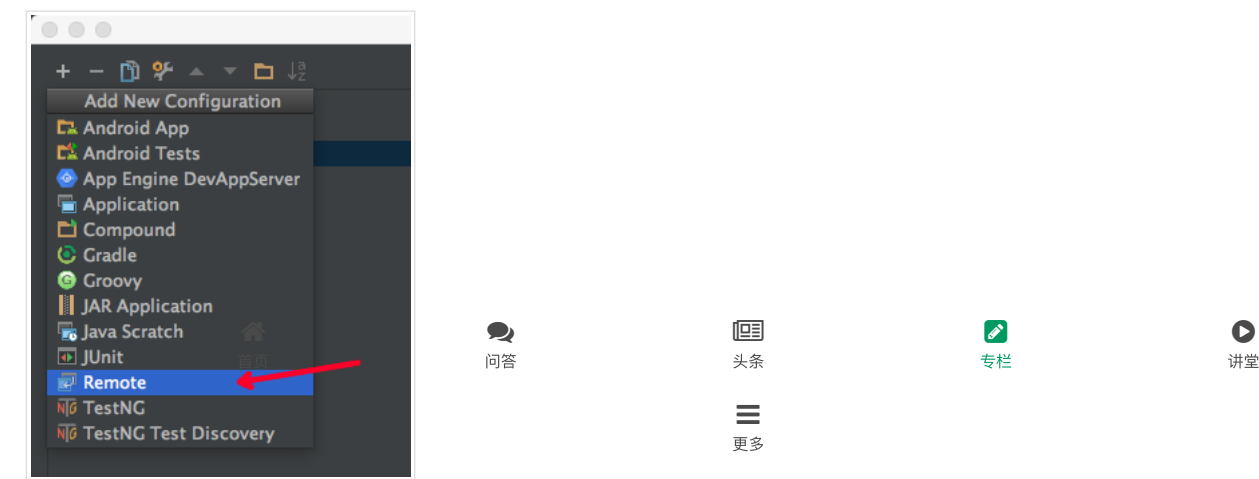


这时任务就不会继续执行了，挂起等待Debugger的handshake!

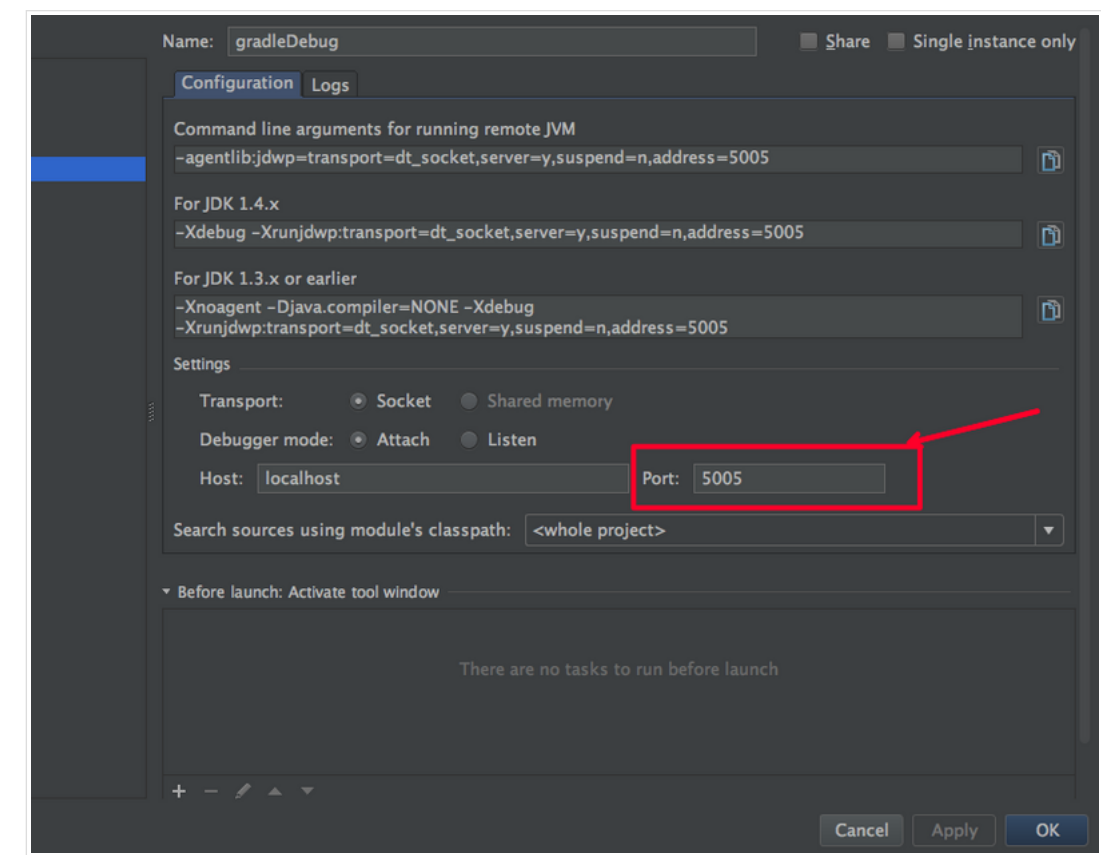
这时我们启动IDE，以IntelliJ为例，打开我们的gradle插件项目（包含正在运行的插件源码的项目），然后找到工具栏的执行区域，进入如下：



在界面里添加一个运行项，Remote：



需要配置的最核心参数就是如图所示的端口号，就是上文中的address：



注意到下面有一个Search sources using ...意思是从什么范围查找源码，保持默认就好了，这个debugger收到JDWP消息后会从当前工程中搜索源码，基于当前的源码进行与gradle命令行中的JDWP agent进行交互。

OK，如果插件源码中打好断点，执行到代码时就会挂起在相应位置了，大部分调试功能都很好使，大家可以尝试一下！这个技巧还是比较简单的，大家只要了解一下JDWP的核心原理，就可以理解上述过程了！

时间有限，如有不妥，欢迎指正！

2017年02月06日发布

...

...